Function Minimization and Automated Alignment of Kirkpatrick-Baez Mirrors

Jerry Lee
Beamline Controls and Data Acquisition Group, Advanced Photon Source Engineering Support Division
Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439. U.S.A.; Contact: lee214@uiuc.edu

Abstract

There are numerous optimization tasks involved with particle accelerator physics. Fitting twiss parameters, maximizing dynamic apertures and reducing the beam's focal spot sizes are but just some of the instances of optimization work being carried out in accelerators. On a broader sense, a large class of problems in many fields of research can be reduced to the issue of finding the smallest value taken on by a function of one or more variable parameters. With that in mind, this paper will be presenting several techniques which can be used to minimize a multivariable function. The underlying motivation for doing so lies in its potential application to the automated alignment of Kirkpatrick-Baez mirrors. Such mirrors are used to focus the X-ray beams produced from synchrotron sources and their alignment can be thought of as minimizing a function of six variables, two for each direction that a mirror can move in. As such, the optimization methods being presented can be used to obtain the most efficient alignment of the Kirkpatrick-Baez mirrors. The main issue that we will have to contend with when mapping the alignment of the mirrors into a response function is that of noise within the function itself. Thus, the optimization techniques will have to be modified to account for this. Also, the motors that control the mirrors can only move in discrete steps. This means that the solutions obtained from the minimization methods may not be implementable. We will discuss strategies to overcome this issue as well.

Introduction

X-ray microscopy is being applied to an increasing number of microstructural problems in materials science, biology, and other disciplines. X-ray crystallography has even been used to characterize the structure of one of the most important protein complexes of the H5N1 virus, the most common strain of bird flu, and this could be critical in obtaining a cure for the deadly disease [1]. As the quality of the X-ray beam is directly related to its focal spot size, it is imperative that the beam be focused to as small an area as possible. One way to focus the beam is with a setup known as the Kirkpatrick-Baez mirror configuration, where two mirrors are placed at glancing angles with respect to the beam. The mirrors are highly polished metallic surfaces, usually coated with platinum, and they are arranged orthogonally with respect to one another so as to successively focus the X-rays in the horizontal and vertical directions. A typical set-up of a Kirkpatrick-Baez mirror focusing system is shown in the diagram below.

By tilting the mirrors in the x, y and z planes, the focal spot size of the X-ray beam can be varied. World-renowned research facilities have reported spot sizes of approximately a micron in diameter. By properly adjusting the position of the mirrors, it has been shown that this value can be obtained for any similar set-up [2]. As such, the problem of minimizing a function of up to six variables, which is the maximum degree of freedom each mirror has, bears significant interest to us. Thus, this paper will investigate the various minimization techniques that can help achieve this goal. Of course, once suitable minimization approaches have been identified, strategies to modify these standard solutions to deal with the issue of noise within the response function, as well as that of the motors controlling the mirrors only being able to move in discrete steps, will be outlined.

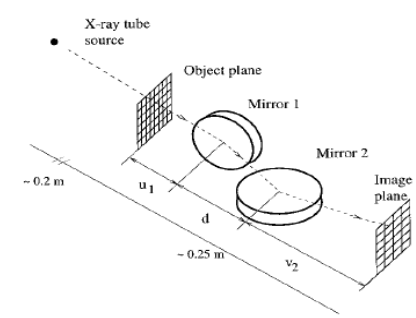

Figure 1. A typical set-up of a Kirkpatrick-Baez mirror focusing system [2]. Note the positions of the orthogonally arranged mirrors. (Picture courtesy of Bakulin)

Grid Search

Initially, we shall be considering functions of just one variable, since almost all problems can be understood most easily in this simplest case and also because some multivariable algorithms contain steps that require one-dimensional minimization techniques. We shall begin our analysis of the various function minimization techniques with the simplest of them all – the grid search method. A grid search consists of choosing equally spaced points within the range of the single parameter to be evaluated, since we are dealing with one-dimensional problems. The function is evaluated at each of the chosen points, and the lowest value found is retained. If the spacing between the points is Δx, then one of the points is sure to be within $\frac{\Delta x}{2}$ of the true minimum, although technically speaking, it may not be the point corresponding to the lowest value. Nonetheless, if the function does not vary too much over distances in the order of Δx, then one would assume that this method gives the minimum within a range of about Δx [3].

The grid search method is absolutely convergent and is stable to all sorts of functions. However, it is extremely inefficient. If the function does not vary too much over a distance of Δx, then many of the function evaluations become unnecessary, especially those in regions where the function value is known to be large. In other words, the algorithm does not take into account what it has learnt about the function. The inefficiency of the grid search method is best illustrated with a numerical example. For instance, if a hundred points are chosen to evaluate each variable of a function, one that contains six variables would require 10^{12} points to resolve. As such, more efficient methods of optimization have to be investigated.

Golden Section Search

The one-dimensional approach to the golden section search starts with three points, x_1, x_2 and x_3; where $x_1 < x_3 < x_2$ and $f(x_3) < f(x_1)$ and $f(x_2)$. The algorithm updates the points so that the middle point has a value that is less than that of the end points. Hence, based on the diagram below, one would try x_4 to get $x_1 < x_3 < x_4$. Subsequently, x_5 would be chosen so that $x_3 < x_5 < x_4$. Then, x_6 would be chosen such that $x_3 < x_6 < x_5$ and so on and so forth.

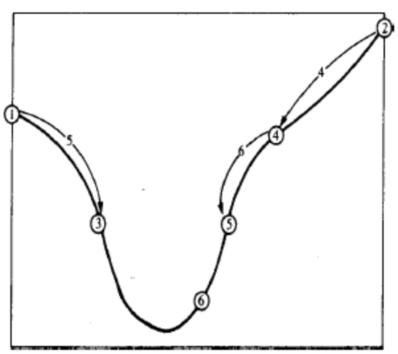

Figure 2. The subsequent points are chosen so that the middle point is always smaller than that of the end points. (Picture courtesy of McGrew)

This minimization technique gets its name from the fact that the subsequent points are always chosen 0.38197 into the larger segment, 0.38197 being the golden ratio. The algorithm is completely robust and its accuracy improves linearly with the number of function evaluations. In addition, even if the function is a smooth one, one would still be able to obtain the minimum of the function, albeit after many iterations.

To counter this effect of requiring many evaluations, Brent's method is used to supplement the golden section search. When used together, the golden section search and Brent's method are perfect for minimizing one-dimensional functions. However, as we will see later, this combinational approach is ineffective when dealing with multivariable functions.

Brent's Method

As mentioned earlier on, the golden section search technique is ineffective in dealing with smooth functions. Brent's method was thus created to overcome this limitation. A caveat to take note of is that one has to actually know that the function is parabolic near the minimum before Brent's method can be used. When this criterion is fulfilled, the algorithm converges very rapidly and is always able to find a minimum.

Brent's algorithm makes use of three points to determine a parabola. Once the pseudo-parabola has been defined, it is used to find a next point. Safety checks are created within the algorithm itself to prevent the parabola from oscillating between two wrong points. This method ensures that a minimum can always be found for functions that are parabolic near the minimum. However, because the algorithm requires such specific conditions for it to work, it is not as robust as other search techniques, in particular the grid search method. Nonetheless, it is useful to be able to call upon it when the golden section search technique fails. When used together, the golden section search and Brent's method are able to minimize virtually every one-dimensional function. Unfortunately, this approach is not very useful when dealing with multivariable functions, as far too many evaluations are required for both search algorithms.

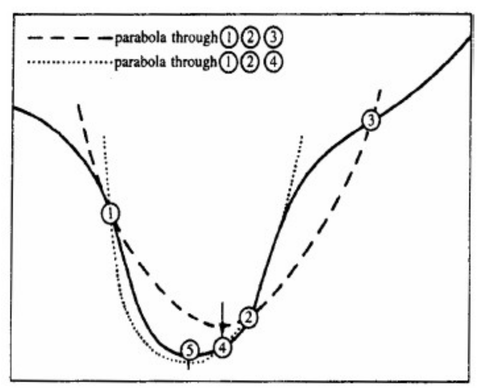

Figure 3. Brent's method chooses the subsequent points by creating a pseudo-parabola that passes through three points of the original function. (Picture courtesy of McGrew)

Stepping Methods in Many Variables

It is a general rule in function minimization that one should not expect good one-dimensional techniques to be effective when extended to higher dimensionality. This is best illustrated by the increase in complexity when applying the grid search approach to multivariable functions. Nonetheless, it is natural for us to try to minimize each variable separately, one after the other. In doing so, we are effectively applying one-dimensional function minimization techniques to each of the variables of a multidimensional function. An example of such an approach would be the single-parameter variation technique, discussed in greater detail below. As we will soon see, such a method is ineffective in dealing with multidimensional functions, as the minimum of a particular variable may not be found at the minimum of another variable.

Single-Parameter Variation

Since the criteria for obtaining a minimum is finding a stationary point, we want to try to make each variable's derivative vanish. The single-parameter variation approach does this in a sequential manner, making each derivative disappear one after the other. Hence, one would seek a minimum for each variable using one of the techniques described earlier on. However, an obvious flaw with this approach is that when one has finished minimizing with respect to x_{i+1}, x_i or earlier variables may no longer be at a minimum, so one would generally have to start all over again. Fortunately, the process does converge, although it may take a long time to do so for functions of many variables [4]. As such, this process is considered too inefficient to be applied to most multivariable functions, and for that reason, we will not be considering it in our implementation.

Simplex Method

To avoid the problem that we faced with the single-parameter variation technique, it might be worthwhile trying to minimize all the variables at once, instead of minimizing one variable at a time. The simplex method is one instance of such an approach. A simplex is a geometric solid in n-dimensions, with n + 1 vertices. Hence, a simplex in two dimensions would be a triangle, while a simplex in three dimensions would be a tetrahedron. The method gets its name from the way the algorithm is carried out. An initial simplex, consisting of n + 1 points, with n being the number of variables in the function to be minimized, is chosen at random. The highest vertex is reflected across the other surface and the process is repeated until the highest point is not decreased. Subsequently, the simplex is shrunk and the highest point is reflected across the other surface again. As such, the minimization algorithm can be thought of as crawling along the function, much like an amoeba would. In fact, some refer to the simplex method as the amoeba technique for this very reason [5]. The figure below best demonstrates this procedure.

There are several benefits to the simplex procedure. Not only is it very easy to implement, one also does not have to assume that the function is smooth for the algorithm is able to handle discontinuous functions with ease. Furthermore, it is easy on the part of the programmer to visualize what is happening, as the function is not minimized along any particular direction. Instead, as mentioned before, the function is minimized vertex by vertex.

Of course, as with all algorithms, there are some limitations as to what the simplex method can accomplish. The simplex technique is not as efficient as other high order routines, which we will discuss later on. Not only does the simplex algorithm require more function evaluations, its precision does not increase very quickly. To further compound matters, the simplex method has a tendency to be tricked by long flat valleys, and the solution obtained would often be far off from the intended target. The simplex technique is also not very good at dealing with narrow valleys as it takes many evaluations to reach the bottom of the valley, where the solution often lies [6]. Nonetheless, the advantages of the simplex procedure far outweigh its deficiencies. Hence, the simplex method is extremely popular as an optimization tool and we will be incorporating this towards the alignment of the Kirkpatrick-Baez mirror configuration.

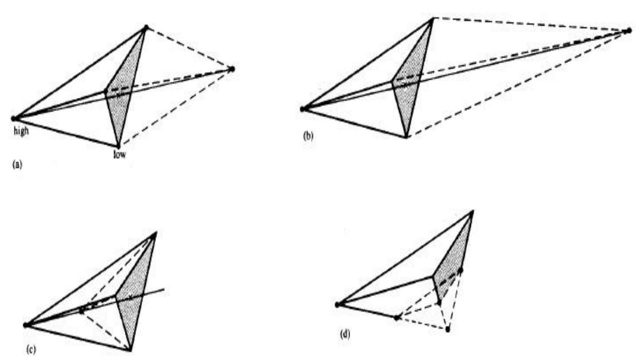

Figure 4. This figure illustrates the simplex procedure. In (a), an initial simplex is picked randomly. Then, the highest vertex is reflected across the other surface as seen in (b). The process is repeated until the highest point is not decreased as illustrated by (c). Finally, in (d), the simplex is shrunk in preparation for the next series of minimization. (Picture courtesy of McGrew)

Steepest Descent

After extolling the benefits of minimizing multivariable functions in its entirety, we return back to our original approach of minimizing a function one variable at a time, albeit with some modifications to the method. Instead of merely tackling the function, more information about the function is used in the solving algorithm. In particular, gradient methods make use of the first derivative of the function in their minimization algorithms. The method of steepest descent is but one instance of such an approach. It is considered a gradient method, as it uses derivatives to predict good trial points that are relatively far away. Do note that this does not necessarily mean that the algorithm is following the gradient, but only that the gradient is used to find the next point in the minimization routine. In essence, the steepest descent approach consists of a series of one-dimensional minimizations, each one along the direction of the local steepest descent, which is the gradient at the point where each new search begins. Naturally, the direction of the gradient would not be constant along a line, even for general quadratic functions, so we would expect many iterative steps to be necessary. Still, the method almost certainly guarantees convergence for quadratic functions, which explains its popularity as a minimization technique.

Since the algorithm involves minimization along each axis in turn, it is apparent that the method requires an auxiliary linear minimization approach. However, according to Murphy's Law of Minimization, function valleys never lie along a primary direction [7]. A method is thus needed to help determine a better direction for minimization. As a result, the conjugate gradient direction search was created with this very purpose in mind.

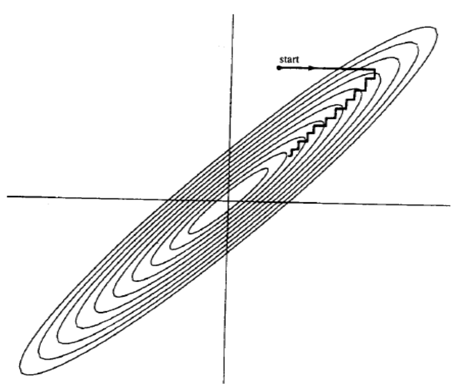

Figure 5. A pictorial representation of the search algorithm used in the steepest descent method. Here, we are dealing with two dimensions, so this gives us steps that look just like the single parameter variation method with the axes rotated in such a way that they line up with the gradient at the start point. An unfortunate limitation of this approach is clearly illustrated here: if each linear minimization is exact, then successive searches must be in orthogonal directions. (Picture courtesy of McGrew)

Conjugate Gradient Direction Search

In the method of conjugate gradients, successive one-dimensional minimizations are performed along conjugate directions with each direction being used only once per iteration. This elegant approach allows the algorithm to converge in as little as two iterations when dealing with quadratic functions. It is also very effective in handling long straight valleys, a condition that the steepest descent method was unable to cope with [8]. Let us briefly explore how the algorithm achieves this. Suppose that the function and its gradient can be evaluated at two different points, $\bar{x}_0$ and $\bar{x}_1$ This would give us the following differences, namely: $\Delta\bar{x} = \bar{x}_1 - \bar{x}_0$ and $\Delta\bar{g} = \bar{g}_1 - \bar{g}_0$ where $\bar{g}_i$ is the gradient at each point, i being either 0 or 1. Then, if the function were quadratic with hessian matrix $\bar{G}$, we would obtain $\Delta\bar{g} = \bar{G}\Delta\bar{x}$. Any vector $\bar{d}$ orthogonal to $\Delta\bar{g}$ would then be conjugate with respect to $\Delta\bar{x}$. This can be expressed as:

$$\bar{d}_1^T\Delta\bar{g} = \bar{d}_1^T\bar{G}\,\Delta\bar{x} = 0,$$

which implies that there must be a method for obtaining conjugate directions without prior knowledge of $\bar{G}$. This technique would of course be based on the change in gradient along a previous direction, which is exactly what the conjugate gradient direction search method does [9].

As mentioned earlier, the conjugate gradient technique utilizes successive one-dimensional minimizations. Unlike the method of steepest descent, the minimizations are performed along conjugate directions rather than along each axis. This difference is what allows the conjugate gradient method to follow long straight valleys. The first step of the algorithm involves choosing an initial direction, $\bar{d}_0$. It is taken as $\bar{d}_0 = -\bar{g}_0$, where $\bar{g}_0$ is the steepest descent vector at $\bar{x}_0$. Next, we let the minimum along this direction be at $\bar{x}_1$ where the gradient is $\bar{g}_1$. Then, the next search direction $\bar{d}_1$, which we want to be conjugate to $\bar{d}_0$, must be a linear combination of the only two vectors we have at hand:

$$\bar{d}_1 = -\bar{g}_1 + b\bar{d}_0,$$

where b is an arbitrary constant.
The conjugacy condition is:

$$\bar{d}_1^T\bar{G}\bar{d}_0 = \bar{d}_1^T\bar{G}(\bar{x}_1 - \bar{x}_0) = 0,$$

or

$$(-\bar{g}_1^T + \mathrm{b}\bar{d}_0^T)\bar{G}\bar{d}_0 = (-\bar{g}_1^T - b\bar{g}_0^T)(\bar{g}_1 - \bar{g}_0) = 0.$$

Since $\bar{x}_1$ is a minimum along the direction as $\bar{d}_0 = -\bar{g}_0$, the direction $\bar{g}_0$ is orthogonal to the gradient at $\bar{x}_1$, so that $\bar{g}_1^T\bar{g}_0 = 0$. We are then left with:

$$b = \frac{g_1^T g_1}{g_0^T g_0}$$

so that the new conjugate direction is

$$\bar{d}_1 = -\bar{g}_1 + \left(\frac{g_1^T g_1}{g_0^T g_0}\right)\bar{d}_0.$$

This process is repeated to generate n directions, each one conjugate to all the others. This simple formula holds for all the successive conjugate directions, so that:

$$\bar{d}_{i+1} = -\bar{g}_{i+1} + \left(\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}\right)\bar{d}_i.$$

The conjugate gradient direction search method is much more robust than the steepest descent approach and is a very useful tool for optimizing functions. However, twisting valleys can cause the conjugate directions being generated to become degenerate. Also, implementing the algorithm is a challenge in itself due to its complexity. Nonetheless, the pros far outweigh the cons and thus, we will be incorporating this towards the alignment of the Kirkpatrick-Baez mirror focusing system.

Implementation towards Alignment of Kirkpatrick-Baez Mirrors

Based on the various optimization techniques outlined above, a suitable program incorporating the best elements of each method needs to be developed. However, as we have seen, no one method can be optimum in the sense that it can be used on all functions. In addition, even for a given function, it is highly unlikely to find a method that works well in all regions, far from the minimum as well as near to it. As such, we will attempt to tailor the program to the function's needs. A decision tree enabling the user to choose a method for his function has been suggested by Fletcher [10]. While this is indeed ingenious, the user has to have some prior knowledge of the function, a luxury one does not have when attempting to align Kirkpatrick-Baez mirrors.

Therefore, we have decided to instead incorporate the golden-section search, Brent's method, the simplex method and the conjugate gradient direction search algorithm into a general program of sorts. The rationale for doing so is because of the inherent advantages each approach brings to the table. When used together, both the golden-section search and Brent's method are able to handle almost any one-dimensional function. The simplex method, together with the conjugate gradient search technique, can handle most multivariable functions. However, both of these multivariable optimization techniques require an auxiliary linear minimization routine, which is exactly what the golden-section search and Brent's method are providing. Hence, if all four routines are used simultaneously, then it is highly likely that they would be able to give a good solution pertaining to the optimum alignment of the mirrors.

When implementing the program, one also has to consider the fact that the movement of mirror is being controlled by three sets of motors. As each motor can only move in discrete steps, the solution obtained from the program may not be implementable. This is a consideration that the programmer has to account for when creating the algorithm. Furthermore, as mentioned earlier on, we will have to contend with the issue of noise within the function itself. The optimization algorithm will thus have to be robust enough to handle any noise in the function values.

Results

It turns out that the program, containing the four optimization techniques, that was described in the previous section is able to handle some noise within functions very well. For the case of the alignment of the Kirkpatrick-Baez mirrors, the program is able to give us good solutions, while taking into account the fact that the motors controlling the mirrors can only move in discrete step sizes. However, the program is unable to handle anything more complicated than parabolas with noise. As one is unable to tell what sorts of functions the alignment of the mirrors can churn up, much less what sort of noise may be produced within the function, it is obvious that more work needs to be done in this field. Still, for the most part, the program appears to work well in that acceptable solutions, close to the minimum, can be obtained for well-behaved parabolic functions.

Conclusions

In summary, various methods for optimizing one-dimensional and multi-dimensional functions have been presented. These techniques have been applied to the problem of aligning a Kirkpatrick-Baez mirror focusing system, an issue that can be thought of as minimizing a function of up to six variables, the maximum degree of freedom that the motors which control the mirrors can move in. As the motors can only move in discrete steps, the minimization methods that were described had to be modified as they could only solve standard functions without any noise in them. In addition, the function describing the mirror configuration is subjected to noise. Thus, any algorithm hoping to obtain the optimum alignment of the mirrors had to account for this as well. A strategy to cope with these two issues was outlined, and a program able to handle noise in parabolic functions, while taking into account the discrete step size consideration, was presented.

Acknowledgements

I would first like to thank my co-advisors, Tim Mooney and Pete Jemian of the Beamline Controls and Data Acquisition Group at the Advanced Photon Source Engineering Support Division, for their invaluable guidance and support, and for agreeing to supervise me throughout this internship. I am also indebted to them for the many discussions about research, academia, and life in general, during my stay at Argonne National Laboratory. Thanks also go out to all personnel involved in the Lee Teng Internship Programme, all of whom I have learnt much from.

References

[1] X. He, J. Zhou, M. Bartlam, R. Zhang, J. Ma, Z. Lou, X. Li, J. Li, A. Joachimiak, Z. Zeng, R. Ge, Z. Rao, and Y. Liu. Crystal structure of the polymerase PA_C–$PB1_N$ complex from an avian influenza H5N1 virus. Nature. DOI10.1038 (2008).

[2] A. Bakulin, S. M. Durbin, T. Jach, and J. Pedulla. Fast imaging of hard x rays with a laboratory microscope. Applied Optics. Vol. 39, No. 19 (2000).

[3] F. James. Intrpretation of the shape of the likelihood function around its minimum. Computer Physics Communications. Vol. 20 (1980).

[4] W. W. Woo, S. A. Svoronos, and O. D. Crisalle. A directional forgetting factor for single-parameter variations. American Control Conference Proceedings. WP5 (1995).

[5] W. Rudin. Principles of mathematical analysis. McGraw-Hill, New York (1976).

[6] R. B. Darst. Introduction to linear programming: applications and extensions. Prentice Hall, New York (2001).

[7] W. H. Press, W. T. Vetterling, S.l A. Teukolsky and B. P. Flannery. Numerical recipes in C: the art of scientific computing. Cambridge University Press, Cambridge (1992).

[8] M. R. Hestenes, E. Stiefel. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards. Vol. 49, No. 6 (1952).

[9] F. James. Function Minimization. CERN Publications, Geneva (1967).

[10] R. Fletcher. Methods for the solution of optimization problems. Computer Physics Communication. Vol. 3 (1972).